REMARKS

The claims are amended to clarify differences between the claimed invention and the art applied by the Examiner. Claims 1-20 remain, with no claim previously allowed.

The Examiner has rejected Claims 1-19 as anticipated by *Schneck et al* (US 5,933,498). The Applicants respectfully traverse this rejection.

(Although this application was filed with 20 claims, the Office Action did not mention or examine Claim 20. The undersigned believes this omission of Claim 20 in the Office Action was an oversight by the Examiner and, accordingly, is not inferring an allowance of Claim 20 from the nonrejection of that claim.)

The present invention addresses and solves a problem different from that of *Schneck* . Accordingly, and not surprisingly, the Applicants have invented and are claiming an invention materially different from what *Schneck* discloses. These differences in problem and in solution are set forth below.

The present invention prevents unauthorized installation of a particular software product by associating a license file with that product. This license is associated with one particular product by adding an Installer Identifier (IID) to the license file. As the specification discloses, the IID may be generated by hashing the files on the media containing a particular software product, or by other means. To install a software product protected accordingly to the present invention, a set-up program will transfer the require run-time files and the license file from the software media to the user's hard drive, i.e., on a local machine. The set-up program also may require that the prospective user enter a product key associated with the software media. The set-up program then generates an IID based on the predetermined characteristics of the media, and checks the generated

IID against the IID stored in the license file on the same media. If the generated IID matches the IID stored on the media license file, this match verifies that the media license file was not tampered with prior to installation, and that the license file was intended for use with the software product on that media. If there is a match, the media license file and a hardware identifier (HWID), identifying the particular local machine on which the software is being installed, are stored in a hardware signature file on that local machine.

At run-time, according to the present invention, the locally-stored license file with the HWID of the local machine is computed and compared with the license in the hardware signature file. If there is a match, the local license file is presumed as properly associated with the installed software product, and an execution of that software is enabled.

The present invention thus restricts the installation of a software product by generating an installer identifier — representing a characteristic of the software product — and comparing that identifier to a installer identifier previously stored on the software media. If the generated installer identifier and the stored installer identifier match, the license file is stored on the local machine and a complete installation of the software product occurs. That stored license file can be subsequently accessed to enable execution of the installed software product on the local machine.

Schneck is styled as a method and device "for controlling access to data". In actuality, that reference concerns controlling access to digital data by encrypting at least portions of the data, preventing access to those protected portions in other than a non-useable form, determining rules for access rights to the data, encrypting those rules, and providing the protected portions of the data and the protected rules. The user can gain

controlled access to the data only in accordance with the rules, but is not allowed to install the data on a local machine. Indeed, *Schneck* teaches that all protected data transferred to the user's machine during execution of a program is destroyed and all files are closed, when execution is concluded (column 19, lines 7-12).

In further detail, *Schneck* discusses and deals with various threats to copying of program material (column 7, line 55-column 8, line 57). These include encrypting or scrambling the output signal to render that signal useless without decryption or unscrambling capability in the "output device" (typically, a TV or VCR), protecting the output signal by making it unavailable outside an "access mechanism", e.g., a sealed-unit computer with tamper detection; preventing digital copying by using a secure coprocessor of signals from the protected program material; and tamper-detection measures that cause the rules, encrypted data, and decrypted protected data to be destroyed in response to attempted circumvention by tampering. In other words, *Schneck* seeks to avoid any installation, i.e., permanent downloading, of a software product on a user's local machine. In contrast, the *present invention* facilitates installation of a software product, albeit in a manner restricting the installed product to a particular license file identified with that product.

The rejection asserts that *Schneck* discloses generating an installer identifier, and comparing that generated installer identifier to a stored installer identifier. However, *Schneck* does not anticipate those elements of the present invention.

According to the rejection, Version Number 127, Authentication 128, and License Number 130 in *Schneck* correspondent to the step of generating an installer identifier. However, *Schneck* discloses that his elements 127, 128, and 130, which are elements of

the structure of rules 116 shown in Fig. 3 (column 10, lines 59-64) actually are authored and included with the packaged data for distribution. Please see column 11, lines 44-67. Those elements thus are not generated by *Schneck* in response to a request to install the software product on a local machine and, indeed, *Schneck* is concerned with *preventing* any installation, as that reference discloses only authorizing *performance* of encrypted program material.

Turning to the step of "comparing the generated installer identifier to a stored installer identifier, columns 17 - 20 of *Schneck* fail to disclose this step. *Schneck* teaches obtaining and decrypting the rules prior to a data access or selection, but those decrypted rules are not compared to the original rules or to any other so-called "installer identifier" stored on the program media. Accordingly, *Schneck* fails to disclosé the step of comparing the generated IID to a stored IID.

The rejection further states that *Schneck* discloses "storing a license in response to a match between the generated installer identifier and the stored installer identifier", asserting that *Schneck's* "Rules" are equivalent to Applicants' license. Assuming *arguendo* the asserted equivalency, *Schneck's* rules are decrypted and stored in the access mechanism 14 in response to an application to access the data according to the stored rules. Please see column 19, lines 12-45 and column 20, lines 6-15. No match between a stored IID and a generated IID is involved in *Schneck* and, as mentioned above, that reference fails to disclose generating an installer identifier and would have no use for one. That reference thus fails to anticipate the step of storing a license file on the local machine in response to match between the generated installer identifier and the stored installer identifier.

Lastly, the rejection asserts that *Schneck* discloses enabling a complete installation of the software product, in response to a match between the generated IID and the stored IID. This step in Claim 1 now calls for enabling a complete installation of the software product on the local machine in response to the match between generated and stored IIDs. However, *Schneck* fails to disclose either the original or the revised wording of this step, as that reference does not enable or desire a complete installation of the program material contained on the software product.

Accordingly, it should be understood that *Schneck* fails to anticipate a method for restricting the installation of a software product onto a local machine, comprising the steps recited in amended Claim 1. Substantially the same steps are present in amended Claim 11, which further requires the step of enabling the execution of the software product on the local machine. The step of "enabling the execution...", added in Claim 11, which also recites "enabling a complete installation of the software product on the local machine...", thus further distinguishes the claimed method from *Schneck*, which enables only execution of selected data, but not installation of a software product on a local machine.

The remaining claims are novel over *Schneck* for the reasons set forth above. In addition, regarding Claims 3 and 13, *Schneck's* rules are formed as part of the process of producing the packaged data (column 11, lines 45-49), and thus are not generated as part of an installation method as in the claimed invention. The same comment applies as well to the rejection of Claims 4 and 14 and to Claims 5 and 15.

As to Claims 6 and 16, *Schneck's* "Authentication (hash) 128" is part of the rules structure 116 (column 10, lines 59-62) authored and stored on the disk or other software

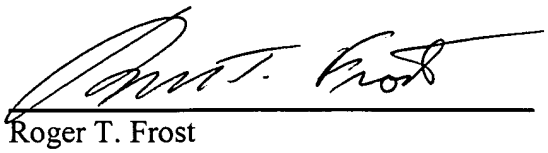
media (column 11, lines 45-54). That authentication (hash) 128 in the reference is not part of a generated installer identifier or of any other value generated in response to a request to install the software product on a local machine, and thus does not meet the limitations added by Claims 6 and 16.

Claim 20 likewise is novel over *Schneck* for the reasons discussed above. In particular, *Schneck* does not disclose a set-up program operative to generate an installer identifier representing a characteristic of a software product media. Moreover, that reference does not disclose a set-up program operative to compare the generated installer identifier to a stored installer identifier on the software product. Further yet, *Schneck* does not disclose enabling the installation of at least one run-time file on a local machine in response to a determination that the generated IID matches the stored IID. Accordingly, Claim 20 defines a system novel over *Schneck.*

The foregoing is submitted as a complete response to the Office Action identified above. The Applicants submit that the present application is in condition for allowance and solicit a notice to that effect.

Respectfully submitted,

MERCHANT & GOULD



Date: June 11, 2004

Roger T. Frost
Reg. No. 22,176

Merchant & Gould, LLC
P.O. Box 2903
Minneapolis, MN 55402-0903
Telephone: 404.954.5100